**Breve Descripción del Problema.**

Se va a realizar un video juego llamado “Space Invaders” donde el jugador moviendo su nave principal horizontalmente en la parte inferior de la pantalla deba disparar a los enemigos que se estarán acercando poco a poco, esto con el fin de que los enemigos no lleguen donde se encuentra la nave. Existirán distintos tipos de hileras de enemigos así como jefes, donde el jugador deberá ir sobrepasando cada uno haciendo que estos nunca lleguen a la parte inferior.

**Historias de Usuario:**

Como jugador quiero:

1. Que la nave se mueva horizontalmente en la parte inferior de la pantalla.
2. Que hayan enemigos en la parte superior de la pantalla.
3. Que los enemigos estén acomodados por hileras los cuáles se muevan hacia abajo y hacia los lados.
4. Que la nave deba eliminar los enemigos mediante disparos
5. Que la nave deba evitar que los enemigos lleguen a la parte inferior de la pantalla.
6. Que existan varios tipos de hileras.
7. Que la velocidad de los enemigos aumente conforme se avanza de nivel.
8. Que cada vez que se destruya un enemigo, estos se desplacen hacia el centro.
9. Que todos los enemigos sean iguales.
10. Que haya un jefe dependiendo el tipo de hilera, para destruir a este se requieren entre 2 y 5 disparos.
11. Que cuando se destruya el jefe se pase al siguiente nivel u otro tome su lugar (dependiendo de la hilera)
12. En la pantalla se deberá indicar el tipo de hilera actual y el próximo por salir, el nivel actual y el puntaje actual.
13. Que en un tipo de hilera los enemigos tengan distintas resistencias.
14. Poder controlar la nave principal con el teclado o mediante el acelerómetro del celular.
15. Que se muestre información del puntaje, tipo de hilera actual y la siguiente por venir en la pantalla del celular.

**Historias de Usuario según Criticalidad y Secuencia de Uso**

**Orden por secuencia de uso:**

1, 2, 4, 3, 5, 8, 9, 6, 10, 11, 7, 13, 14, 15

A continuación se mostrará una matriz para asignar la criticalidad y secuencia de uso que van a tener las tareas establecidas en cada bloque de historias de uso. La clasificación de los colores para la matriz se explicará a continuación:

1. Color Rojo: representa el nivel de prioridad “Alto”, por lo que debe realizarse con mayor urgencia, y se debe considerar muy importante para el desarrollo del proyecto.
2. Color Amarillo: representa el nivel de prioridad “Medio”, por lo que es poco urgente pero importante.
3. Color Verde: representa el nivel de prioridad “Bajo”, por lo que es poco urgente y no tan importante.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Secuencia de Uso | | | |
| Criticalidad |  |  |  |  |
| 1, 2, 4 | 3, 5 |  |
|
| ↑ Minimal System Span ↑ | | | |
|  |  | 6, 8, 10, 7 | 11 |
|
|  |  | 9 | 13, 14, 15 |
|

Tabla 1. Matriz de Criticalidad y Secuencia de Uso de las Historias de Usuario

**Plan de Iteraciones**

**Cronograma y Descomposición de Historias de Usuario en Tareas.**

**Bloque 1.**

- Investigación e implementación de la librería gráfica LibGDX.

**Bloque 2.**

- Búsqueda de los sprites y el escenario para todos los elementos gráficos del juego.

- Hacer que la nave principal aparezca sobre el escenario.

- Hacer que la nave se mueva horizontalmente.

**Bloque 3.**

- Hacer que los enemigos aparezcan en la parte superior.

- Hacer que los enemigos tenga sus sprites en movimiento.

- Hacer que los enemigos se muevan hacía los lados y hacía abajo.

**Bloque 4.**

- Hacer que la nave dispare.

- Crear la lista simple con sus respectivos métodos.

- Crear el tipo de hilera básico.

- Cuando los enemigos son golpeados con un disparo que estos desaparezcan.

- Los enemigos se desplazan hacia el centro cuando se elimina uno de ellos.

**Bloque 5.**

- Crear el tipo de hilera A.

- Crear el jefe (aparece siempre en una posición aleatoria) y que este al ser eliminado se gane la ronda.

**Bloque 6.**

- Crear la lista doblemente enlazada con sus respectivos métodos.

- Crear el tipo de hilera B (El jefe cambia de posición aleatoria con los demás enemigos).

**Bloque 7.**

- Crear la lista circular con sus respectivos métodos.

- Crear el tipo de hilera C (Un enemigo aleatorio obtiene la posición de jefe cada que este es eliminado).

**Bloque 8.**

- Investigar sobre la implementación de “bubble sort”.

- Crear el tipo de hilera D (todos los enemigos tienen distintas resistencias, ordenados de izquierda a derecha, de menor a mayor resistencia).

**Bloque 9.**

- Crear la lista circular doblemente enlazada con sus respectivos métodos.

- Crear el tipo de hilera E (el jefe siempre está en el centro y la hilera gira en sentido reloj).

**Bloque 10.**

- Cuando los enemigos lleguen a la nave del jugador sea “game over”.

- Crear los distintos niveles.

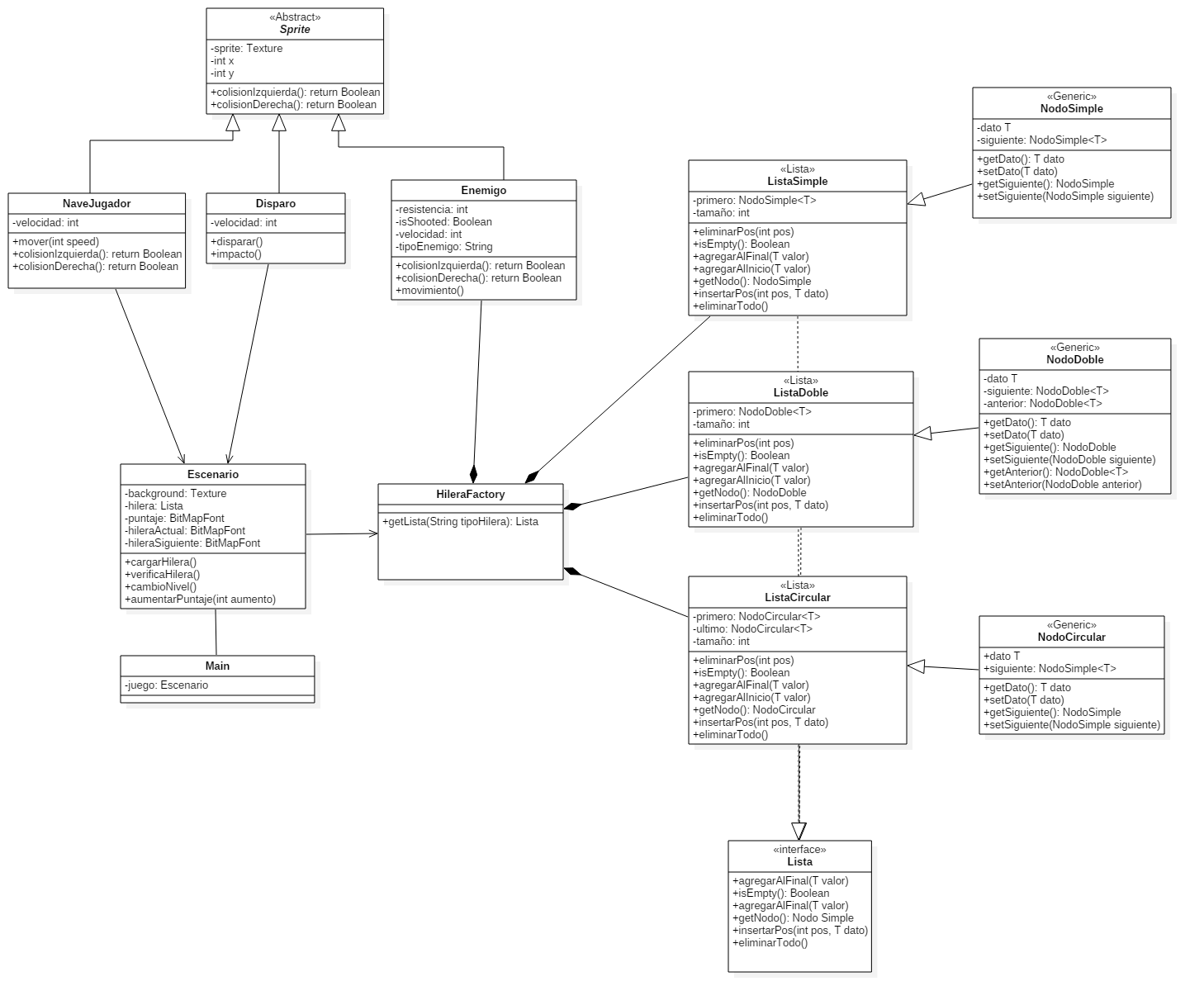
- Que las hileras vengan al azar en cada nivel.

- Mostrar en pantalla el puntaje obtenido, el tipo de hilera actual y el tipo de hilera siguiente.

**Bloque 11.**

**-** Realizar la aplicación para el celular de manera que esta pueda controlar la nave mediante el acelerómetro.

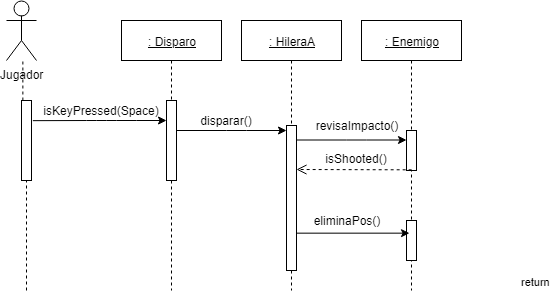
**Diagrama de Clases Inicial**



**Diagramas de Secuencia**

**Historia de Usuario #4:**

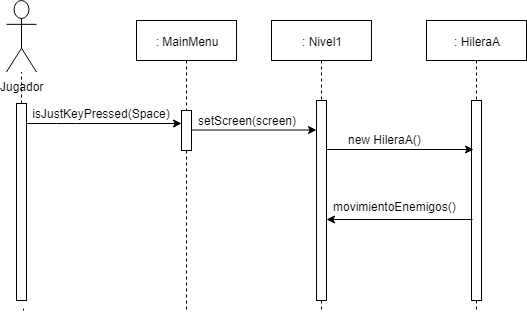
- Que la nave deba eliminar los enemigos mediante disparos.



El jugador presiona la tecla “Espacio” y este hace que el disparo se acomode justo frente de la nave y se mueva en dirección vertical hacia arriba, se realiza la validación si alguna nave fue impactada, si lo fue se cambia su atributo “shooted” a true y se procederá a eliminarse de la lista.

**Historia de Usuario #4:**

-Que los enemigos estén acomodados por hileras los cuáles se muevan hacia abajo y hacia los lados.



El jugador entra y se topa con el menú donde si presiona la tecla “Space” entrará al nivel 1, este crea una nueva hilera y al ser creada se ejecuta el método que se encarga del movimiento hacia abajo y hacia los lados del enemigo.